

The DØ Trigger Examine

Elliott Cheu, *University of Arizona, Tucson, Arizona*

Abstract

This note describes the Trigger Examine package used for monitoring the online trigger components. This package runs automatically at the start of each run and produces histograms that the shift captain can monitor for problems with the data taking. This package also copies the histograms to a web-accessible area so that anyone with a web browser can check the detector status.

Contents

1	Introduction	1
2	Scripts	1
3	TrigSimCert Process	2
3.1	Building TrigSimCert	3
4	TrigCertAna Process	4
4.1	Building TrigCertAna	5
5	The gmpotter Process	5
5.1	Reference Plots	6
6	Summary	6

1 Introduction

The trigger examine (TrigEx) consists of four main components: start up scripts, framework job, histogram macros, and histogram plotters. A flow chart of the trigger examine can be seen in Figure 1. The examine is controlled by a single script that runs continuously to monitor the state of the run and launch the appropriate scripts and processes depending upon whether there is an ongoing run or not. The framework job is launched by this script at the start of each new run. The trigger examine currently runs on d0ol75 and reads a fraction of the events from the monitoring stream to produce a single .root file with data from all of the D0 trigger components. Simultaneously, a second process is launched that reads in the .root file from the framework job to produce a new .root file with the appropriate histograms that need to be monitored. Finally, a third process called gmplotter is also started. This process is part of the gmbrowser package,[1] and it generates .gif files of pages of plots to be monitored. These .gif files can be viewed with any web browser, allowing one to easily get a snapshot of the detector performance, with regards to the L1, L2 and L3 triggers.

2 Scripts

The scripts needed to run the trigger examine are part of the trigger_examine package. In fact, the trigger examine package only consists of these scripts now. Within this package there are the following scripts:

- run_TrigExAuto.py: This script is the main script that starts everything. The script first determines a few settings and then instantiates TrigExAuto and runs it. Within global monitoring, the script is run by the d0gm account and can be started (or restarted) with the following set of commands:

```
setup global_mon  
gmresetTrigExAuto
```

These commands can be started from any online machine, but typically we have been using d0ol68 as our main console. There are two ways to check the status of this process.

- Open a web browser and go to <http://www-d0online.fnal.gov/groups/gm/GM.html>
- Start gmgui (setup global_mon; gmgui)

A process called TrigExAuto is displayed on the screens of either the gmgui or the web-based display. A red or green dot indicates the status of the process. If the process is not running, using the gmresetTrigExAuto command should restart it.

A log file of the TrigExAuto process (and most of the following scripts) can be found in \$GM_LOG_DIR. This environment variable is defined when you setup global_mon. The run_TrigExAuto.py script makes use of the following scripts.

- TrigExAuto.py: This script does all of the work. It first determines whether there is a current store and run. If a valid store and run exist, then the process checks to see whether the TrigSimCert, TrigCertAna and the gmpotter process are running. If so, the process sleeps for a bit before checking again. If any of these process are not running, then the TrigExAuto.py script restarts the appropriate process. If the TrigExAuto.py process does not detect both a valid store and run, then it sleeps for a period before checking again. Currently, if there is no store, the process waits five minutes, while if it's in store (but not in a run), the process waits for two minutes. The following scripts are imported and used by the TrigExAuto.py script.
 - * TrigExAuto_params.py: Defines various parameters used by TrigExAuto.
 - * SESgetinfo.py: Communicates with SES to get the store and run state.
 - * TriggerExamine.py: Launches the various trigger examine processes by calling trigger_examine.sh.
- trigger_examine.sh: This script is used to start the three main trigger examine processes. It first checks to see if the TrigSimCert process is already running. If so, it moves on to the next task. If this process is not running, then it starts the TrigSimCert process. The trigger_examine.sh script then kills the histogram macro process, TrigCertAna, if it exists. It then starts a new TrigCertAna process. The script does the same for the gmpotter process. In principle, if these last two processes are already running, there is no reason to kill them and start them again. However, there is no harm in doing so, since neither process requires a lot of CPU time and they quickly catch up to the current event. The trigger_examine.sh script calls the following three scripts to start the corresponding processes.
 - run_trigsimcert_dist.job: TrigSimCert
 - runroot.job: TrigCertAna
 - run_gmpotter.job: gmpotter

3 TrigSimCert Process

The TrigSimCert[2] process is a framework job that runs, via the distributor, on raw data events. Within the framework, this job is running the trigsimcert package.

To get trigsimcert to attach to the distributor a few slight modifications have to be done before building this executable. These modifications are documented in the HOWTO file in the trigger_examine package. However, they are repeated below. The code for the trigger examine process is kept in the /online/examines/⟨version⟩/trigger_examine directory, where ⟨version⟩ is a production version. The current production version used by the trigger examine is p20.08.01.

When it is running, the TrigSimCert process connects to the distributor. The information required for this connection is contained in the file: /online/data/d0online/d0online.names.py. This information is parsed by the run_trigsimcert_dist.job which passes the information to TrigSimCert.

After it starts running, TrigSimCert creates a run directory in /scratch2/trigger_examine.⟨version⟩/rundir. The last nine run directories are stored as rundir_1 through rundir_9 to archive recent runs for debugging purposes. Then, TrigSimCert changes to the run directory and writes the log and .root files here.

During a run approximately 40K events will be processed. Because of a feature in root, if the output .root file exceeds a few GB, root will automatically close the output file. This will cause TrigSimCert to exit. At this point the TrigExAuto process will note that there is no framework job running and restart it. This effectively limits the length of time that each TrigSimCert process can run to approximately 4 hours.

3.1 Building TrigSimCert

To build the framework job, you can do the following:

```
* setup D0Run <version>
* setup d0cvs
* newrel -t <version> trigger_examine
* cd trigger_examine
* d0setwa
```

```
Get one of the latest versions of read_event_daq,
* addpkg -h read_event_daq
* edit read_event_daq/rcp/ReadEventDaq.rcp and set debug_level=0
```

```
Add in trigsimcert
* addpkg trigsimcert
* Modify bin/OBJECTS to include RegReadEventDaq
* gmake all
```

```
Add the trigger_examine package to get the appropriate scripts.
* addpkg -h trigger_examine (There isn't really anything in this package)
```

Modify `$GLOBAL_MON_DIR/ups/global_mon.table` to set `$GM_TRIGEX_VERS` to current D0 version.

Changing the value of `$GLOBAL_MON_DIR` is important because it tells the TrigExAuto script which version of the trigger examine to run. You can directly modify the `global_mon.table` file with your favorite editor. However, you have to be logged on as `d0run` to modify it. Also, once you have changed the version you should do the following:

```
setup global_mon      (This will pick up the new version number)
gmresetTrigExAuto     (This will restart TrigExAuto with the new number)
```

If you use the above procedure, the next time there is a new run, the trigger examine will start with the updated version. If you want the TrigSimCert process to start immediately with the new version, then you just have to kill the TrigSimCert process after restarting TrigExAuto.

4 TrigCertAna Process

The TrigCertAna executable resides in the `trigsimcert/macros` directory. The main program is contained in the file `TrigCertAna.cpp` while the code for each subsystem is located in the `macros/HistoFill` directory.

The purpose of this executable is to read the data stored in the `trigsimcert.root` file and generate separate histograms for each plot of interest. When this executable starts, it reads the event information to determine the current run number. The run number is stored in the output file as the histogram `event_HISTO1`. Then, TrigCertAna opens an output file of the form `TrigEx_run<run number>.root`. This file is located in `/scratch2/output/trigger_examine` on `d0ol75`. The process also creates a soft link called `TrigEx.root` in the same directory that points to the current `.root` file. This is used by both `gmbrowser` and `gmplotter` to determine the latest trigger examine `.root` files.

The macro process loops over all events stored in the `trigsimcert.root` file. When it reaches the last event in the input file, it sleeps for a few seconds, before rereading the event information. If it finds that the number of events has increased, TrigCertAna then starts reading the input file again from the previous event up to the maximum events that now exist in the file. At the end of each event loop, the histograms are written to the output file, and the event loop repeats. The histograms are written using the `TObject::kOverwrite` flag. This causes the previous instance of the histogram to be overwritten so that the browsers will see updated plots during each pass. At times the TrigCertAna process will be writing at the same time as the browser is trying to read a histogram. This can cause some communication

problems that result in the plot not being properly plotted. However, this will clear up on the next pass.

The TrigCertAna process does not know about the end of the run. This is why when the trigger_examine.sh script starts, it kills all running TrigCertAna processes and starts a new one.

4.1 Building TrigCertAna

TrigCertAna has to be built after the framework job, TrigSimCert, has been built. This is because the macro process needs to use the same dictionaries that are built when creating the framework process. The TrigCertAna process is built in the following manner:

```
* setup D0RunII <version>
* cd trigsimcert/macros
* make
```

5 The gmplotter Process

Pages of related plots are created using the gmbrowser display program. The gmbrowser configuration file is called TrigEx.cfg and it resides in the d0gm/global_mon/config directory. More information about gmbrowser can be found in D0 Note 5323. To start the gmbrowser window in the control room, one can issue the command

```
gmstartTrigEx_browser
```

This command can be issued from any d0olXX machine. The preferred machine is d0ol68.

A program related to gmbrowser is called gmplotter. This program reads in the same configuration file as gmbrowser. It is meant to run as an autonomous program and generate .gif files that can be viewed from a web browser. The gmplotter program is started by the run_gmplotter.job program. It can also be started by hand by typing:

```
gmstartTrigEx_plotter
```

This program loops over all of the defined histogram pages and creates a single .gif file for each defined page in TrigEx.cfg. The output of this program is written to: /mnt/www/htdocs/groups/gm/TrigEx In this same directory is a file: TrigEx_plots.php. This script is executed whenever one visits the appropriate web page (see above).

In addition, the gmplotter process also creates a single postscript file of the form: TrigEx_run<run number> in the directory d0gm/global_mon/plots/TrigEx. The location of this directory is specified in the TrigEx.cfg file with the parameter GM-Plot.PS.Dir. The postscript file is used as an archive of the trigger examine output.

Of course, one can always rerun gmbrowser on a specific TrigEx.root file. These files are stored on the /scratch2 disk and we have output files going back to Feb. 2005. To view a specific postscript file, the alias gmview has been created. So, one can do:

```
gmview <TrigEx/PhysEx> <run number>
```

5.1 Reference Plots

The gmplotter and gmbrowser programs both use reference histograms to help guide the shifter for the appropriate ranges of histogram values. Recently, we have placed two reference histograms on each plot for low and high luminosity running. The two histograms are specified in TrigEx.cfg by the parameters:

```
Root.Ref.File:  
Root.Ref.File2:
```

Currently, the first parameter is used to display the low luminosity plot, while the second shows the high luminosity plot. To generate these histograms we have been running trigsimcert offline over a number of raw input files. However, one could also just copy the appropriate files from the /scratch2/output/trigger_examine directory. These files are the output files from the TrigCertAna process.

After putting new reference files in the appropriate directory, one does NOT need to restart anything. Both gmbrowser and gmplotter reopen the reference histograms whenever they plot a page. While this may seem somewhat inefficient, it is necessary because of how network files are handled by root.

Descriptions of the plots used by the trigger examine can be found in [2] and also online at: <http://www-d0online.fnal.gov/groups/gm/GM.html>

6 Summary

We have built a relatively robust system for monitoring data from the L1/L2/L3 detectors. This system starts and stops automatically in parallel with the global run. The output of this system can be viewed locally in the control room, or more widely via the web, providing an easy way to determine whether or not problems exist with the online data taking.

References

- [1] E. Cheu, H. Prosper, D0 Note 5323.
- [2] C. Belanger-Champagne, Y. Coadou, D. O'Neil, Steve Beale, D0 Note 4570.

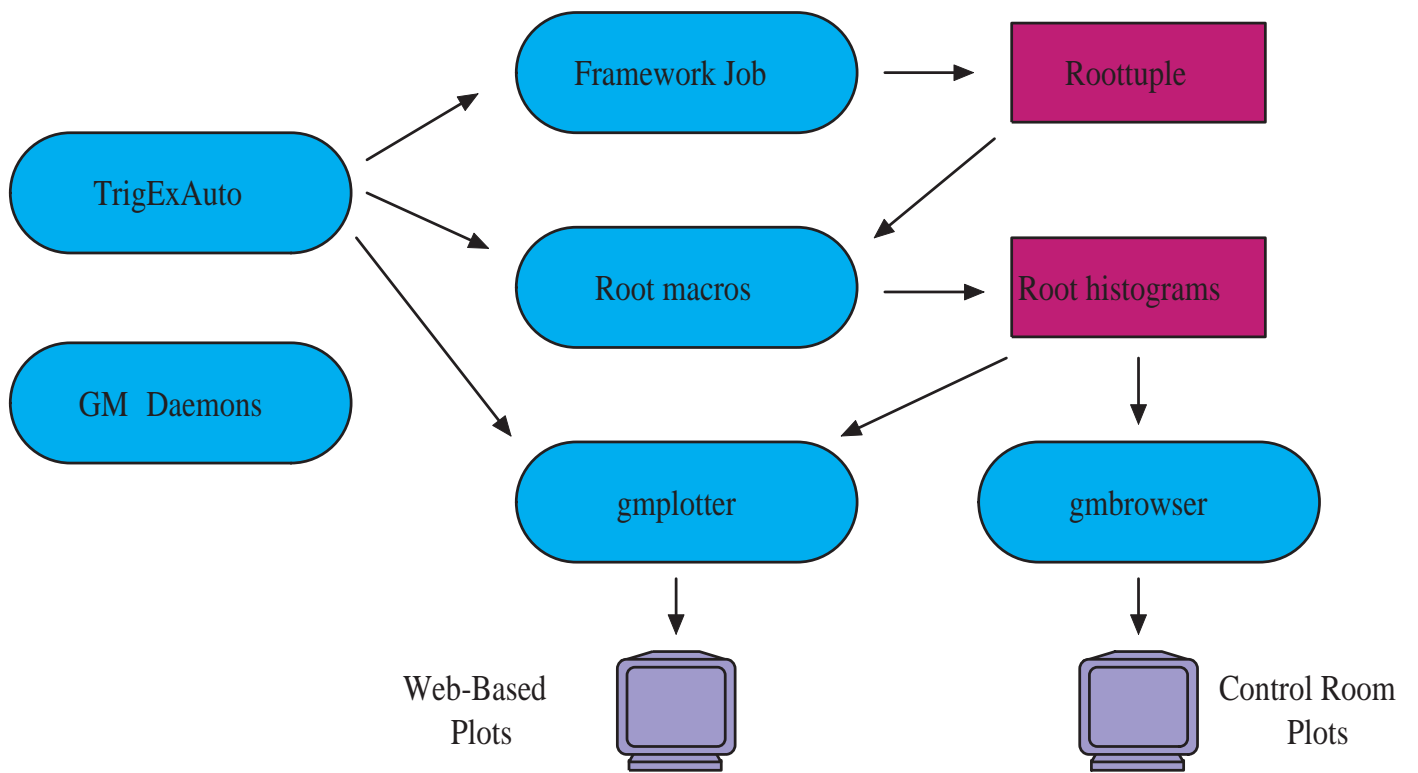


Figure 1: Trigger examine flow chart.